Articulatory feature recognition using dynamic Bayesian networks

J. Frankel, M. Wester, and S. King. Articulatory feature recognition using dynamic Bayesian networks. Computer Speech & Language, 21(4):620-640, October 2007.

Articulatory feature recognition using dynamic Bayesian networks

Joe Frankel, Mirjam Wester and Simon King

Centre for Speech Technology Research
University of Edinburgh
2 Buccleuch Place
Edinburgh EH8 9LW, UK
Tel: +44 131 651 1769
Fax: +44 131 650 4587

Abstract

We describe a dynamic Bayesian network for articulatory feature recognition. The model is intended to be a component of a speech recognizer that avoids the problems of conventional "beads-on-a-string" phoneme-based models. We demonstrate that the model gives superior recognition of articulatory features from the speech signal compared with a state-of-the art neural network system. We also introduce a training algorithm that offers two major advances: it does not require time-aligned feature labels and it allows the model to learn a set of asynchronous feature changes in a data-driven manner.

Key words:
Articulatory feature recognition, dynamic Bayesian network, neural network, automatic speech recognition, virtual evidence

1 Introduction

The majority of automatic speech recognition (ASR) systems describe speech as a linear sequence of phones. Words are generated from models that are concatenations of phone models. The notion that a word is realized as a sequence of

Email address: joe@cstr.ed.ac.uk (Joe Frankel, Mirjam Wester and Simon King).
URL: http://www.cstr.ed.ac.uk/∼joe/ (Joe Frankel, Mirjam Wester and Simon King).

non-overlapping phone segments, i.e. the "beads-on-a-string" paradigm (Ostendorf, 1999), makes it extremely difficult to model the variation that is present in spontaneous, conversational speech. This variation in part arises from the overlapping, asynchronous nature of the articulators which together form the speech production mechanism. Conventional systems use context-dependent phone models to deal with this variation, although this can only deal with certain effects (Jurafsky et al., 2001). It also forces the introduction of parameter sharing schemes to alleviate problems of overparameterisation and consequent data sparsity.

Many researchers have argued that better results could be achieved by *modelling* the underlying processes of co-articulation and assimilation, rather than simply *describing* their effects on the speech signal.

Systems which take this explicit articulatory modelling [1] approach are diverse in the types of data and types of models used. The articulatory representation can be continuous-valued, including parameters derived from measured articulation (Zlokarnik, 1995; Wrench, 2001; Frankel, 2003) and continuous-valued production-*inspired* parameters (Nix and Hogden, 1998; Richards and Bridle, 1999). Discrete representations include quantized measured articulator positions (Stephenson et al., 2000; Markov et al., 2003) and features derived from existing (e.g., phonetic) labels according to linguistic knowledge (Kirchhoff, 1999; King and Taylor, 2000; Wester et al., 2001; Metze and Waibel, 2002).

It is the latter type with which we are concerned in this article: articulatory features (AFs). We describe a system which follows on from work by King and Taylor (2000), and introduce two important advances. First, we use a dynamic Bayesian network, which enables modelling of inter-feature dependencies whilst allowing asynchrony between features, and provides a powerful framework within which the model can be extended to word recognition. Second, we use an embedded training scheme to go beyond the limitations of time-aligned phonetically-transcribed data; this scheme is also important for training asynchronous models.

1.1 Articulatory features

The definition of a set of articulatory features draws inspiration both from the theory of distinctive features (Chomsky and Halle, 1968), in which phonemes are represented in terms of binary features (for example voiced/voiceless), and the gestural theory of speech production (Browman and Goldstein, 1992) which describes speech production in terms of a set of articulators working in loose synchrony. A set of articulatory features suitable for ASR must:

[1] Within which we include models of the physical articulators as well as the use of more abstract representation of articulation, such as features.

feature	values	cardinality
manner	approximant, fricative, nasal, stop, vowel, silence	6
place	labiodental, dental, alveolar, velar, high, mid, low, silence	8
voicing	voiced, voiceless, silence	3
rounding	rounded, unrounded, nil, silence	4
front-back	front, central, back, nil, silence	5
static	static, dynamic, silence	3

Table 1
Specification of the multi-valued articulatory features used in this work.

- correlate with the acoustic signal, to allow automatic recognition using models learned from data
- provide a compact representation, to limit the number of parameters required in the model used to recognize the features
- encode the distinctions necessary for eventual word discrimination

We have chosen to use a set of multi-valued features along dimensions including *voicing*, *manner of articulation*, and *place of articulation*. The feature sets used in this work are shown in Table 1 and described more fully in Section 2. In addition, the core terminology used in this paper is explained in Table 2.

Our ultimate goal is a speech recognizer which uses an internal articulatory feature (rather than phoneme) representation: AFs will mediate between words and acoustic observations. Deriving the internal representation from an articulatory, rather than phonetic, perspective will allow explicit modelling of co-articulation, and hence a parsimonious encoding of the contextual variation characteristics of natural speech.

In this article, we focus on the task of articulatory feature recognition, i.e. automatic assignment of AF labels to unseen speech. A variety of modelling approaches have been proposed in previous studies, including artificial neural networks (ANNs) (Kirchhoff, 1999; King and Taylor, 2000; Chang et al., 2001; Wester et al., 2001), hidden Markov models (HMMs) (Kirchhoff, 1999; Eide, 2001; Metze and Waibel, 2002), support vector machines (SVMs) (Niyogi et al., 1999; Juneja and Espy-Wilson, 2003; Scharenborg et al., 2006) and dynamic Bayesian networks (DBNs) (Stephenson et al., 2000; Livescu et al., 2003; Frankel et al., 2004; Wester et al., 2004; Frankel and King, 2005). These studies have highlighted some of the useful properties of articulatory features, such as reliable recovery from acoustic parameters, noise-robustness (Kirchhoff, 1999), and increased language-independence

Term	Meaning
articulatory feature *or* feature *or* AF	one of the six multi-valued articulatory features from table 1; e.g., *manner*
cardinality	the number of possible values a feature can take; e.g., 6
feature value	one of the possible values for a feature; e.g., *fricative*
frame-level	corresponding to individual acoustic observations; i.e., every 10ms
feature value label *or* label	a temporal region where a feature has a constant feature value; c.f. a time-aligned word or phone label
feature classification	determining, for each frame, the most likely feature value for each feature
feature recognition	determining, for an utterance, the most likely label sequence and segmentation for each feature

Table 2
Terminology used in this paper

(compared to phonemes) (Wester et al., 2001).

1.2 Modelling inter-feature dependencies

A key design choice in developing a model for articulatory feature recognition is whether to assume features are statistically independent, or to train a model for simultaneous recognition of all features. Previous approaches have typically assumed independence. For example, Kirchhoff (1999) proposed that separate models be trained to classify the values taken by each feature group, as the complexity of the individual classifiers will be less than that of a monolithic classifier. Assuming that the individual classifiers can be combined effectively, such an approach should lead to improved robustness. Kirchhoff (1999) shows this to be true using word recognition experiments with a hybrid ANN/HMM approach. The baseline system used artificial neural networks (ANN) which were trained to produce phone-class posteriors. A second, feature-based, approach was used in which the output of a set of 5 independent feature classifier ANNs was combined to give phone-class posteriors via a further ANN. The feature-layer system was found to be more robust to additive noise.

Despite the practical benefits of treating features as independent, dependencies clearly exist. For example, Wester et al. (2001) and Chang et al. (2001) have shown that *place* of articulation classification can be improved by training *manner*-value-

specific models.

However, modelling inter-feature dependencies raises the question of how to deal with asynchrony, especially if training on feature value labels which are derived from a time-aligned phone transcription. In a maximum likelihood setting, only feature value combinations which appear in the training data will accumulate non-zero probability mass, so that the model is constrained only to allow the combinations found in the original labels. In the limit, this would reduce to phone recognition with a distributed representation, in which case a major advantage of representing each feature with its own state stream, that of modelling asynchrony between features, is lost.

In Frankel et al. (2004), we proposed using DBNs for articulatory feature recognition. We showed that by modelling dependencies between feature groups, recognition accuracy is increased over an equivalent system in which features are assumed independent. However, the system in Frankel et al. (2004) was trained on phone-derived feature value labels, leading to an overly strong set of constraints on feature co-occurrence as discussed above.

In this paper, we extend our DBN approach to articulatory feature recognition in two directions. Firstly, we present an embedded training scheme designed to learn a set of asynchronous feature changes where supported in the data. In this way we alleviate the problems inherent in training on phone-derived feature labels. Secondly, we show how the DBN framework can be used to enhance the performance of artificial neural network (ANN) AF recognizers, which to date have been the most accurate articulatory feature recognizers.

The remainder of the paper is organized as follows: Section 2 describes the data set and how articulatory features are derived from phonetic transcriptions; in this work, we pay extra attention to diacritics. Section 3 describes an ANN feature classification system, which provides a state-of-the art baseline, and the DBN system is described in Section 4. This is followed by an account of the embedded training scheme and the classification and recognition results for a variety of DBN systems. Section 5 presents an analysis of the asynchrony learned by the models and Section 6 summarizes the contributions of this article.

2 Data

Our experimental work uses data from the Numbers corpus (Cole et al., 1995), a collection of naturally spoken numbers collected at the Center for Spoken Language Understanding (CSLU) at OGI. The utterances were taken from other CSLU telephone speech data collections, and include isolated digit strings, continuous digit strings and ordinal/cardinal numbers. Each utterance in the Numbers corpus

set	utterances	phones	frames
train	10 251	207 728	2 166 806
validation	3 518	72 698	75 9117
test	3 553	73 280	78 7513

Table 3
Division of the OGI Numbers 30-word subset into train, validation and test sets.

has been orthographically and phonetically transcribed following the CSLU Labelling Conventions (Lander, 1997). The set used in this study (Table 3) was selected to contain only the 30 most frequent words and no utterances with truncated words (Mariéthoz and Bengio, 2004).

We chose to work with OGI Numbers because of the availability of detailed phone-level transcriptions which include diacritics (see Section 2.2). A further consideration is that the small vocabulary will be convenient in future work developing a word recognition system.

2.1 From phonetic to articulatory feature transcriptions

In all experiments, the acoustic waveforms are parameterized as 12 Mel-frequency cepstral coefficients and energy, with analysis every 10ms within 25ms windows. The 1^{st} and 2^{nd} derivatives were then appended to give 39-dimensional acoustic observation vectors.

Frame-level feature value labels were generated from the existing time-aligned phone labels using a set of phone-to-feature-value mapping rules. The feature specifications are similar to those used in Frankel et al. (2004) and Wester (2003) with modifications to take into account the findings in (Wester et al., 2004) and to account for the specific characteristics of spoken numbers. For instance, there are a number of phonemes which do not occur in spoken numbers, and therefore appear very rarely in the OGI Numbers corpus (e.g., /b/, /m/, and /h/). As a consequence, some places of articulation occur very infrequently in the data. To avoid problems of data sparsity, labial frames were relabelled as labiodental and all glottal frames relabelled as velar. The findings in Wester et al. (2004) led to the following changes in the mapping rules. The *front-back* feature was enriched by the inclusion of a *central* value and diphthongs were split into two vowels, corresponding to their start and end points, which were each mapped to AFs as usual.

The features, their sets of possible values and cardinalities (number of possible values) are listed in Table 1. For the full table of phone-to-feature mappings, see Appendix A.

diacritic	description	number of occurrences			effect
		train	validation	test	
_*	waveform cut off	207	83	67	–
_?	glottalized	1268	434	408	–
_?*	glottal onset	217	59	75	–
_(	flapped (consonant)	379	131	131	–
_0	voiceless	178	74	77	voiced → voiceless
_:	lengthened	13	2	3	–
_h	aspirated	277	81	67	–
_j	palatalized	6	2	2	–
_r	retroflexion	97	34	44	–
_v	voiced	167	73	49	voiceless → voiced
_w	more rounded	24	12	11	unrounded → rounded
_F	fricated stop	18	6	10	stop → fricative
_~	nasalized	33	12	11	vowel → nasal
_n	nasal release	0	0	2	–
_x	centralized	167	45	57	front/back → central

Table 4
Specification of diacritics encountered in the OGI Numbers data sets, their frequency counts and the effect we determine them to have on the feature value labels.

2.2 Use of diacritic markers

One of the reasons for using OGI Numbers is the careful phonetic transcription, which includes a number of diacritics. Diacritics add fine detail to the base symbol (Lander, 1997). The underlying assumption is that a more detailed phone transcription should lead to a better AF description. Table 4 shows the diacritics that occur in OGI Numbers, their frequency counts in each of the data sets and the effect of each diacritic on the phone-to-feature mapping.

It can be seen in Table 4 that not all diacritics result in changes in the feature value labelling. This is for one of the following reasons:

- the diacritic does not occur frequently enough (e.g., palatalization),
- the diacritic describes a process that would involve adding a new feature value (e.g., flapping would require the addition of a new value to *manner* of articulation)

- the diacritic describes a process that would involve adding a new feature (e.g., lengthening would require the addition of a new feature, *length*).

A set of ANN feature classification experiments on half of the data listed in Table 3 were used to guide the construction of the feature specification. Table 5 shows test set frame-level accuracies for three different mapping strategies. These results clearly show that incorporating *all* diacritic markers leads to lower performance, because mapping glottalization to silence is inappropriate. When glottalization is disregarded, including diacritics in the feature specification leads to similar results as when not including diacritics. It seems a phonetic transcription including diacritics is less beneficial to feature classification than anticipated.

Although diacritics are not very common (only 0.3% of phone mappings are affected) and do not influence feature classification a great deal we nevertheless chose to include them using the "+dia -glot" strategy from Table 5 because of the potential benefits in future work on modelling pronunciation variation. The mapping strategy used in the remainder of this article for the full data set (Table 3) therefore interprets the diacritics as shown in the right-most column of Table 4, which disregards the glottalization diacritic.

frame-level accuracy (%)			
strategy:	-dia	+dia -glot	+dia
see:	Appendix A	Appendix A plus Table 4	Appendix A plus Table 4 plus glottalized phones → silence
manner	**87.2**	87.0	86.3
place	**84.4**	**84.4**	84.0
voicing	**90.8**	**90.8**	90.4
rounding	**86.9**	86.8	86.8
front-back	86.5	**86.6**	86.3
static	**86.9**	**86.9**	86.6

Table 5
Accuracies for an ANN system trained and tested on half of the data from Table 3, using feature value labels derived by various phone-to-feature mapping strategies: without diacritics (-dia), with diacritics except for glottalization (+dia -glot), and with all diacritics (+dia) in which glottalized phones were mapped to silence. The highest accuracy per feature is shown in bold face.

2.3 Evaluating performance

A problem with the task of articulatory feature recognition is performance evaluation. One possibility is to compare AF recognition results using the percentage of frames correct, averaged over all features, and the percentage of frames in which all features are correct together. These measures inevitably compare recognition output against phone-derived feature value labels and so have the drawback of penalizing asynchronous feature-value changes.

Therefore, we additionally give results using the standard recognition measure (usually used for words) of accuracy (in percent):

$$100 \times \left(n(\text{correct labels}) - n(\text{inserted labels})\right) / n(\text{total labels}) \qquad (1)$$

where $n(\cdot)$ indicates "number of" and errors are computed with respect to phone-derived feature value labels. We use the HTK tool HResults (Young et al., 2002).

Using accuracy (which does not take segmentation timing into account) does not penalize asynchronous feature-value changes, but still has the capacity to penalize some of the events we would wish to capture, such as where assimilation should lead to the deletion of a feature value label.

An ad-hoc indication of the degree of coupling between the feature streams, and hence the degree of asynchrony, is given by counting the number of unique combinations found in the decoder output. This measure will be useful in comparing model characteristics before and after embedded training. It is expected that learning asynchronous changes between streams will lead to a greater number of combinations.

3 Articulatory feature classification using artificial neural networks

In order to assess how well our DBN models perform, we need a state-of-the-art baseline with which to compare DBN performance. To this end, a set of artificial neural networks was trained (one for each feature) using the NICO Toolkit (Ström, 1997), an ANN toolkit designed and optimized for speech recognition applications. All networks are recurrent time-delay neural networks, consisting of three layers: an input layer, a single hidden layer, and an output layer with a softmax activation function (similar to the architecture described in Ström (1997)). A 1-of-N encoding of feature values is used for the ANN outputs, so that each network output unit corresponds to one of the discrete values taken by the feature.

During training, input-output pairs consist of a frame of acoustic features and the

phone-derived articulatory feature value labels for that frame.

During testing, for a given acoustic frame, the set of output activations of the network for a particular feature can – after suitable normalization – be interpreted as a posterior probability distribution over the possible values that this feature can take (Table 1). When evaluating the network classification accuracy (i.e., at the frame level), the feature value with the highest activation is chosen.

3.1 Previously-reported results

Table 6 gives results reported in the literature for frame-level AF classification using a similar multi-valued feature system to that used in the present study. The results in Table 6 show consistency despite relating to a variety of databases and phone-to-feature mappings.

frame-level accuracy (%)						
reference	corpus	feature				
		manner	*place*	*voicing*	*rounding*	*front -back*
Kirchhoff (1999)	OGI Numbers	82.0	77.2	89.1	83.2	83.0
Kirchhoff (1999)	Verbmobil	81.5	69.7	87.4	83.3	81.4
King and Taylor (2000)	TIMIT	87	72	93	92	84
Wester et al. (2001)	VIOS	84.9	75.9	88.9	83.2	83.0
Chang (2002)	NTIMIT	85.0	71.2	88.9	82.9	80.9
Wester (2003)	TIMIT	87.0	78.3	92.9	90.6	86.4
Hacioglu et al. (2004)	TIMIT	86.5	73.0	-	-	-

Table 6
Previously reported articulatory feature classification accuracies.

3.2 ANN configuration and tuning

Various configurations and parameter settings for the ANNs were tested on the validation set. These affect the speed of training and final accuracy of the system and include factors such as the size of the hidden layer, the size of the input context window, connectivity between the various layers and the gain and momentum parameters. An exhaustive study of the effect of all these parameters was beyond the scope of this study. Instead, these parameters were chosen on the basis of a combination of previous work (Ström, 1997; Stephenson, 1998; Wester, 2003) and a

limited degree of tuning (only the sizes of the hidden layers were varied) to optimize for the current database.

	feature					
	manner	*place*	*voicing*	*rounding*	*front-back*	*static*
output units	6	8	3	4	5	3
hidden units	300	300	100	200	250	150
connections	65,328	66,360	18,834	41,693	53,462	30,176
validation accuracy	88.3%	85.7%	91.7%	88.1%	87.2%	88.2%
test accuracy	88.5%	85.9%	91.8%	88.3%	87.4%	88.4%

Table 7
ANN frame-level feature classification accuracy on the OGI Numbers validation and test sets as given in Table 3.

The optimum number of hidden units was found to be roughly proportional to the cardinality of the corresponding feature, as shown in Table 7 along with the number of connections (i.e., trainable weights).

For each feature, the feature value with highest associated posterior (i.e., the output with the largest activation from the ANN for that feature) at each frame is compared with the phone-derived feature values to calculate a frame-level classification accuracy. Our validation and full test accuracies are also given in Table 7, and show that our ANN system gives sufficient accuracy to be considered state-of-the-art (c.f. Table 6). The classification accuracy for the place feature is higher than reported in many studies. This may be attributable to our place feature having lower cardinality (which is appropriate for the numbers domain) than in other studies.

The frame-level accuracy averaged across all features is 88.4%, with all feature correct together in 77.5% of frames. The high proportion of frames in which all features are correct together shows that there is a correlation in the distribution of errors between different networks.

4 Feature classification and recognition using dynamic Bayesian networks

Our description of the feature recognition DBNs will proceed as follows. Section 4.1 gives an introduction to the properties of DBNs, which is followed by a description of our feature recognition DBN in Section 4.2. We give details of the embedded training scheme which allows the model to learn asynchronous feature value changes in Sections 4.3 and 4.4 before presenting results in Section 4.5.

4.1 Dynamic Bayesian networks

A Bayesian network (BN) is a particular form of graphical model in which a set of random variables (RVs) and their inter-dependencies are represented as the nodes and edges of a directed acyclic graph (DAG). When depicting Bayesian networks, we follow the notational convention that round/square nodes show continuous/discrete variables, and shaded/unshaded distinguishes observed/hidden variables. For example, in Figure 1, Y is an observed RV, and F^1, F^2, F^3 are hidden and discrete-valued.

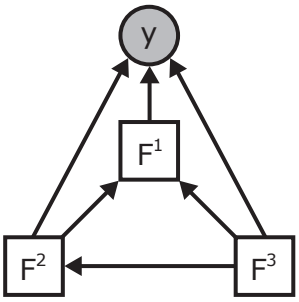

Fig. 1. *Example Bayesian network, in which the joint probability of* Y, F^1, F^2, F^3 *can be factored as* $p(Y, F^1, F^2, F^3) = p(Y|F^1, F^2, F^3)P(F^1|F^2, F^3)P(F^2|F^3)P(F^3)$

The set of edges and their directions in Figure 1 immediately tell us that the joint probability of Y, F^1, F^2, F^3 can be factored as

$$p(Y, F^1, F^2, F^3) = p(Y|F^1, F^2, F^3)P(F^1|F^2, F^3)P(F^2|F^3)P(F^3) \qquad (2)$$

A key concept in this factorization, and one we refer to later, is that of *parents*, denoted $Pa()$. For node Y, we define the $Pa(Y)$ to be the set of nodes on which Y is conditioned, so in the example of Figure 1, $Pa(Y) = \{F^1, F^2, F^3\}$.

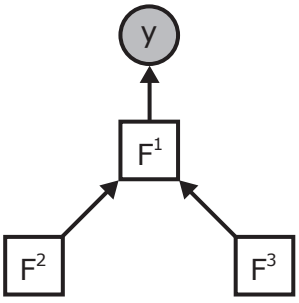

Fig. 2. *Example Bayesian network, in which the joint probability of* Y, F^1, F^2, F^3 *can be factored as* $p(Y = y, F^1, F^2, F^3) = p(Y = y|F^1)P(F^1|F^2, F^3)P(F^2)P(F^3)$

A Bayesian network exploits missing edges (which imply conditional independence) to factor the joint distribution of all random variables into a set of simpler probability distributions. In Figure 2, a number of independence assumptions have been made, for example Y no longer depends directly on F^2 and F^3, and $Pa(Y) = \{F^1\}$. The joint distribution of Y, F^1, F^2, F^3 can now be factored as

$$p(Y = y, F^1, F^2, F^3) = p(Y = y|F^1)P(F^1|F^2, F^3)P(F^2)P(F^3) \qquad (3)$$

Since speech utterances give rise to variable length sequences of data, our model will in fact be a Bayesian network that dynamically "unrolls" to fit the observation sequences. This type of BN is known as a *dynamic* Bayesian network (DBN), and consists of instances of a Bayesian network repeated over time, with additional arcs added to join variables at differing times.

In addition to the notational conventions introduced above, temporal arcs, which join RVs at differing times, are shown with "drop shadows" for legibility. Where appropriate, the decoding version of each model is shown; the training version is typically identical, except for additional variables to encode the known label sequence for the training data.

The DBN notational conventions do not show the nature of the dependencies or the distribution form of the random variables (RVs). We use GMTK (Bilmes, 2002a) to implement our models, in which conditional probability tables (CPTs) are used to describe the distributions of discrete variables given their parents, and Gaussian mixture models (GMMs) are supported for specifying distributions over continuous-valued variables [2] . A further option (described in Section 4.2.2 below) is to incorporate virtual evidence (VE) in the form of ANN posteriors.

Dynamic Bayesian networks form a large class of models which includes the HMM as one special case. DBNs provide an ideal framework to combine information from multiple sources, and offer the potential to build more structured models of the parameterized speech signal than HMMs. In recent years there has been steadily growing interest in the application of DBNs to ASR, aided by the emergence of toolkits such as GMTK. An overview of the field of ASR from a graphical models perspective can be found in Bilmes (2003), and a more theoretical treatment of Bayesian networks can be found in Jensen (2001).

4.2 Feature recognition DBN

The feature recognition DBN can be thought of as being comprised of two components:

[2] Our models do not use hidden continuous variables or continuous-valued parents. At the present time these are not supported by GMTK

Observation process This is the method by which evidence from the acoustic parameters is incorporated into the model. Two possibilities are presented and compared in this work. The first uses the likelihood from Gaussian mixture models (GMMs) (see Section 4.2.1), and the second is by incorporating virtual evidence (VE) in the form of ANN posteriors (see Section 4.2.2) .

State process Discrete random variables encode F_t^k for each feature $k = 1, \ldots, 6$ at time t. During training, these state streams are observed directly or constrained to follow a particular sequence of values (which is derived from the labels). The state is hidden during recognition, which consists of a Viterbi search to find the most likely sequence of values for each state stream.

Deriving a joint observation process for all features poses a number of difficulties, which we illustrate with the example of GMMs in Section 4.2.1 below. In this work, we suggest that coupling through the *state* process, rather than the observation process, will provide the desired combination of distributing the modelling among a set of low-cardinality classifiers, one for each feature, whilst also giving the benefits of modelling the dependencies between features. This is described in more detail in Section 4.2.3.

4.2.1 Observation process - Gaussian mixture model (GMM)

In order to condition the distribution of Y_t, the observation at time t, on $F_t^1, \ldots, F_t^6$, the feature variables at time t, we need an estimate of the following conditional Gaussian: $p(Y_t|F_t^1, \ldots, F_t^6)$. As discussed in Livescu et al. (2003), specifying such a distribution – i.e., a Gaussian or GMM for every combination of feature values – leads to problems of data sparsity. Furthermore, since the model will initially be trained using phone-derived feature value labels, there will only be training data available for feature value combinations that correspond to phones.

Given that parameter tying for a feature-based observation process is an open research question, we follow the approach presented in Livescu et al. (2003) and adopt a factored observation model. For each of the six features $F^1, \ldots, F^6$, a conditional GMM is trained (i.e., one GMM for each value that F^k can take). Therefore:

$$p(Y_t|F_t^1, \ldots, F_t^6) = \frac{1}{Z(Y_t)} \prod_{k=1}^{6} p(Y_t|F_t^k) \tag{4}$$

where $Z(Y_t)$ is a normalization constant. With $f^{k,l}$ representing a particular value of the k^{th} feature, the probability density function (pdf) of $p(Y|F_t^k = f^{k,l})$ is modelled by a GMM whose parameters depend only on k and l and do not vary with t. The total number of GMMs required is therefore 29, the sum of the cardinalities of individual features from Table 1, and all use diagonal covariance matrices.

We refer to the DBN with a Gaussian mixture model observation process as a GMM/DBN. Such a model is shown in Figure 3. The round shaded nodes represent

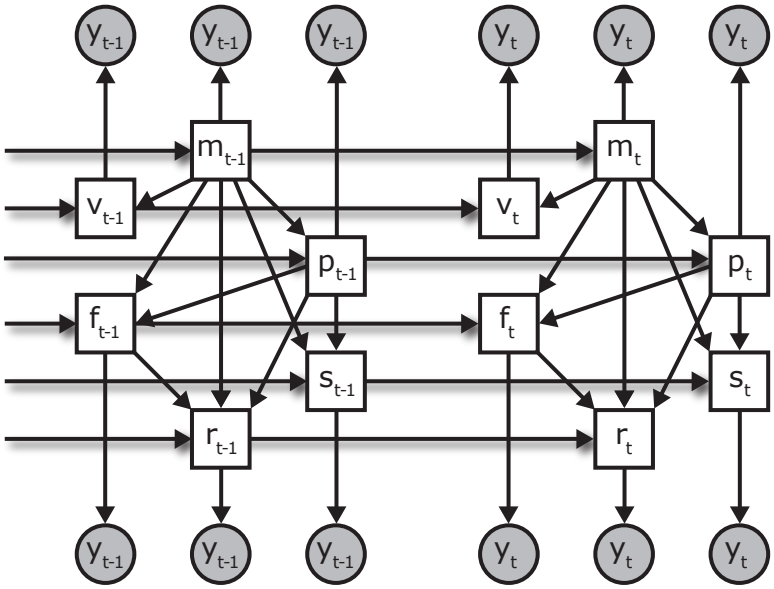

Fig. 3. *Decoding graph for the GMM/DBN which includes dependencies between features. For clarity, the feature variables at time* t *are labelled* $m_t, p_t, v_t, r_t, f_t, s_t$ *(*manner*,* place*,* voicing*,* rounding*,* front-back*,* static*) rather than the abstract names* $F_t^1, \ldots, F_t^6$.

the continuous-valued observed random variables of the observation process. There is one such node per feature, and each one is conditioned on the state of the feature to which it belongs.

4.2.2 Observation process - virtual evidence (VE)

Rather than observing a frame of acoustic parameters Y_t, and using a GMM to assign a likelihood $p(Y_t|F_t^k)$, we can instead incorporate a discrete probability distribution over the values l of feature F_t^k. This probability distribution, which is read in at each frame rather than calculated, is referred to as virtual evidence (VE). Incorporating VE adds the flexibility to include information from any model which can be trained to assign a discrete probability function. Here we incorporate VE from the ANNs described in Section 3.

The hybrid ANN/HMM (Morgan and Bourlard, 1995; Robinson et al., 2002) approach to speech recognition can be expressed as a DBN which incorporates VE. We follow the ANN/HMM example and insert VE as a scaled likelihood calculated as follows. The posterior $p_k(F_t^k = f^{k,l} | Y_t) = p_k(f^{k,l} | Y_t)$ associated with each level l of feature F^k is related to a generative likelihood by Bayes rule:

$$p_k(f^{k,l} | Y_t) = \frac{p_k(Y_t | f^{k,l})\, p_k(f^{k,l})}{p_k(Y_t)} \tag{5}$$

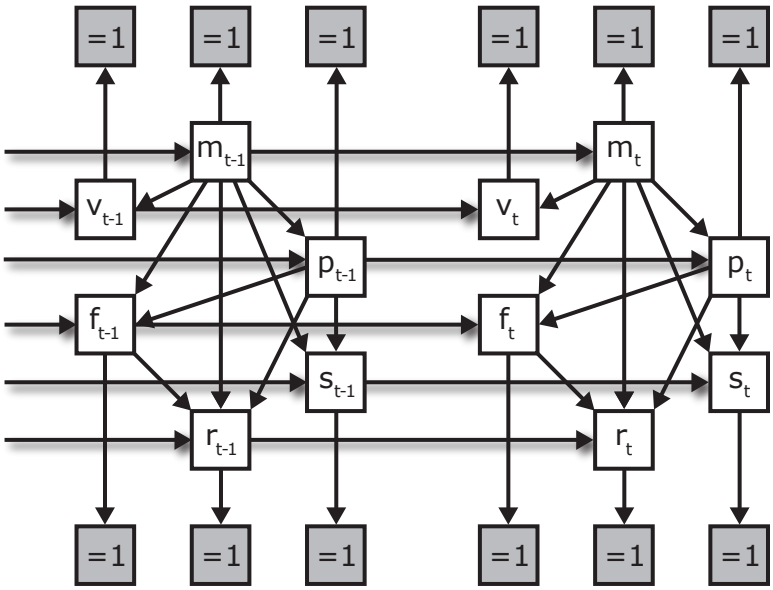

Fig. 4. *Decoding graph for the ANN/DBN which includes dependencies between features and virtual evidence. Notation is the same as Figure 3*

Ignoring $p_k(Y_t)$, which is independent of the feature state, the scaled likelihood is given as:

$$f_k(Y_t \mid f^{k,l}) \propto \frac{p_k(f^{k,l} \mid Y_t)}{p_k(f^{k,l})} \tag{6}$$

We refer to this model, which is shown in Figure 4 as an ANN/DBN. The virtual evidence is depicted and implemented by including a binary discrete random variable V_t^k which is conditioned on the state for feature $k = 1, \ldots, 6$, and always observed to be equal to 1. The VE is then incorporating by reading in $P(V_t^k = 1 | f^{k,l}) \propto f_k(Y_t \mid F^{k,l})$ at each time t.

4.2.3 State process

A special case of DBN state process is one where features are assumed independent, and are only conditioned on their value at the previous time. This amounts to a set of HMMs operating in parallel, and is referred to as a GMM/HMM or ANN/HMM dependent on the type of observation process. An ANN/HMM is shown in Figure 5 and is used to provide a baseline in assessing the contribution of the set of inter-feature dependencies.

We take the view that the strong assumption of independence made in the observation process will be mitigated by retaining a set of interdependencies between features in the state process. These dependencies are encoded using conditional

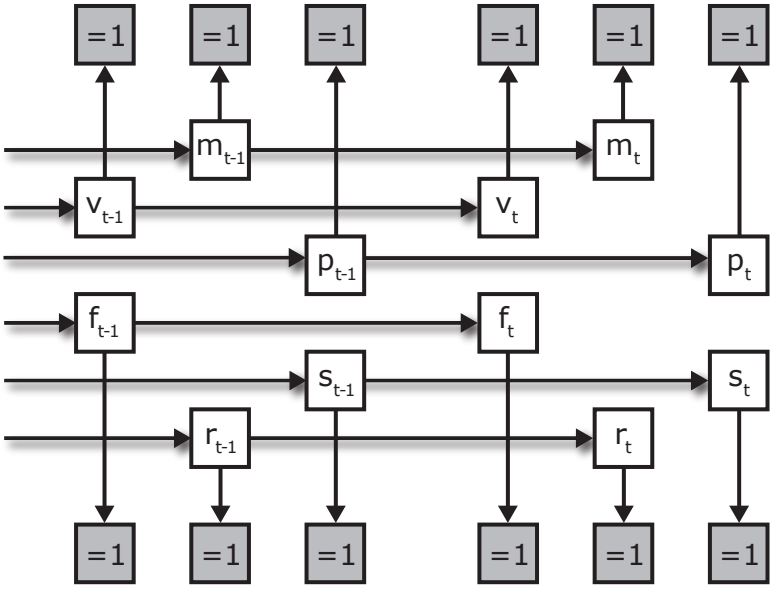

Fig. 5. *Decoding graph for the ANN/HMM in which features are modelled as independent, and virtual evidence is used for the observation process. Notation is the same as Figure 3*

probability tables (CPT), which hold $P(F^{k,l}|Pa(F^k))$ the probability of finding each value l of feature $k = 1, \ldots, 6$ given the values taken by its parents.

The state topology we use is shown in Figures 3 and 4. For details of the method by which this particular set of dependencies was determined, we direct the reader to Frankel et al. (2004). This set of dependencies, based on linguistic knowledge, was shown to give improved AF recognition compared to an equivalent model in which features were modelled as independent. The manner feature can be seen as central to the model, as it is parent to all other features.

The effective size of the hidden state space of this model is the product of the individual feature cardinalities, which gives a total of $6 \times 8 \times 3 \times 4 \times 5 \times 3 = 8640$ possible feature value combinations. As shown in Table 1, each of the features includes a silence level. An addition to our state process, which for clarity we omit from the diagrams, is that we force the use of the silence level to be synchronized between features. A hidden discrete silence/non-silence node is added, and all features are forced to take silence/non-silence values according to its value. This reduces the

number of possible feature combinations to $5 \times 7 \times 2 \times 3 \times 4 \times 2 + 1 = 1681$. All DBNs in the current work incorporate this "synchronized-silence" mechanism.

4.2.4 Summary of the models used in this work

We now give a brief summary of the models for which we present experimental results.

GMM/DBN This is the model shown in Figure 3, in which a GMM observation process is combined with a state in which dependencies are modelled between features. This model is implemented using GMTK.

ANN/HMM This model has a virtual evidence observation process, and a state process in which features are modelled as independent. Unlike other experiments presented in this work, the ANN/HMM models were implemented using the NOWAY start-synchronous decoder (Renals and Hochberg, 1995), which was designed for ANN/HMM speech recognition. The implementation has identical structure as in the GMTK-based experiments, with features modelled using a set of one-state HMMs (one per feature value) connected in a "phone loop", with ANN outputs providing scaled likelihoods. As in the DBN systems, feature transition probabilities were estimated on the train set.

Each of the features was decoded in isolation, and errors across the different features were all counted equally when compiling overall results.

ANN/DBN This is the model shown in Figure 4, in which a VE observation process is combined with a state in which dependencies are modelled between features. This model is implemented using GMTK.

4.3 Method for training feature recognition DBNs

The algorithm we use for training the feature-recognition DBNs has three phases. In the first, models are trained with the phone-derived feature labels observed. During the subsequent steps, embedded training is used in which the sequence of feature value labels is known but the boundary times are not.

4.3.1 Phase 1: training with observed frame-level feature value labels

In the first phase, the phone-derived feature value labels are used as observed values for the feature RVs in the DBN. After training, the conditional probability tables (CPTs) which describe the dependencies between features are extremely sparse (i.e. most entries are zero), and it is the sparse structure of the CPTs which dictates which features values can co-occur and which cannot.

It is important to clarify the role that the conditional independence assumptions

play. For feature F^k, it is the combinations of $F^k \cup Pa(F^k)$ observed in the training data which create the sparse structure in the CPT representing F^k. For example, Figures 3 and 4 show that the *voicing* feature is conditioned on its own value at time $t-1$ and *manner* at time t, and is independent of the values of the other features. It is apparent then that even when the model is trained only on observed phone-derived labels, there is the possibility of inferring combinations of all 6 features which did not occur in the original labelling, although the combinations of each feature and its parents ($F^k \cup Pa(F^k), k = 1, \ldots, 6$) must have been seen in order to accumulate non-zero probability mass in the corresponding entries of the CPTs.

For the GMM/DBN model, the observation process GMMs are estimated during Phase 1 using a process of splitting and vanishing following a scheme adapted from that described in Bilmes (2002b).

4.3.2 Phase 2: embedded training of asynchronous feature CPTs

As discussed above, training on phone-derived feature labels leads to a strong set of constraints on feature co-occurrence. After Phase 1 training, only those CPT entries corresponding to feature value combinations occurring in the training data have accumulated probability mass. In order to learn which additional CPT entries should be non-zero, an embedded training scheme is applied. Our goal is to train a system which is able to model phonetic detail, such as asynchronous feature changes, yet retains sufficient sparsity in the CPTs so that unlikely or impossible feature value combinations are omitted and, as a result, inference can be performed on the model at a practical speed.

The essence of the scheme is as follows: zero-valued CPT entries from the model trained in Phase 1 are raised to some small value, the CPTs renormalized, and then updated using embedded training in which feature value combinations with strong acoustic likelihood accumulate probability mass, giving rise to significant CPT entries, whilst combinations with low acoustic likelihood result in zero or very low probabilities.

In the experiments reported below, the zero CPT cells are raised to $1/(\alpha \text{ card}(F^k))$ where $\text{card}(F^k)$ denotes the cardinality of feature F^k. A value of $\alpha = 10^5$ was used, giving $\frac{1}{\alpha}$ an order of magnitude lower than the smallest CPT entry found after Phase 1 training. Our implementation differed slightly according to the type of observation process, and we discuss each in turn below.

GMM/DBN EM training involves a summation over all allowable state sequences, which proved to be computationally infeasible for the 6-factorial GMM/DBN during the first embedded iteration, even with aggressive levels of pruning. This arises as the feature CPTs have no zero entries, so that all 1681 combinations have non-zero probability. With change-points no longer derived from phone transitions and common across features, the effective hidden state space is pro-

hibitively large. We therefore employ a cascaded approach. Initially, a single iteration of full embedded training is performed for the *manner* and *voicing* features together. *Manner* was chosen to be trained first because it is the parent node to all other features, and *voicing* was chosen because of its low cardinality. Features are then added back into the model one at a time in the order *place*, *front-back*, *rounding* and *static* so that parent variables' CPTs are updated prior to those of their children. A single iteration of embedded training is performed after adding each feature, and only the state parameters are updated at the end of each iteration. By the end of Phase 2, asynchronous CPTs have been learned for each feature, though the observation process parameters are unchanged.

ANN/DBN By contrast, full embedded training of all feature states together was possible for the ANN/DBN. We attribute this to the inherently discriminative nature of ANN posteriors, which gives rise to distributions with relatively low entropy compared to those from GMMs which are trained as generative models. At each frame, the probability mass for each feature tends to be dominated by a single value, with the effect that beam pruning can be used very effectively to reduce the search space, whilst retaining promising hypotheses.

4.3.3 Phase 3: all-parameter update

During Phase 3, all parameters, state and observation, are updated given the asynchronous CPTs computed in phase 2.

GMM/DBN For the GMM/DBN, this phase consists of full expectation maximization (EM) embedded training of state and observation parameters, with the sequence of feature values but not timing enforced. Training continues until convergence which is considered to be when likelihood changed by less than 0.2% between iterations.

ANN/DBN For the ANN/DBN, we first realign the training data using the model after Phase 2 training, then use the new alignments as targets with which to retrain the ANNs. The new set of ANNs are used to calculate scaled likelihoods during decoding, and the state process parameters are trained until convergence as above.

4.4 Training schemes

The training schemes used in the experiments reported in this article are listed in Table 8 and described below. They are made up of combinations of the three phases described above. The key difference between the A and B schemes is whether or not training Phase 2 is included, in which the asynchronous feature CPTs are learned and the dependence on phone-derived feature labels is reduced. Both are split into parts 1 and 2.

scheme	possible feature combinations	training phases (Section 4.3)	GMM/DBN	ANN/HMM	ANN/DBN
A.1	phone-derived	1	√	√	√
A.2		1,3	√		
B.1	any	1, 2	√		√
B.2		1, 2, 3	√		√

Table 8
Summary of the different training schemes, and models to which they are applied.

4.4.1 Training scheme A

Scheme A.1 serves as the baseline: the factored hidden state of the model can only occupy those parts of feature-value-space that correspond to combinations seen during training. The feature value label boundary times are fixed to those of the phone labels. This is the only training scheme applied to the ANN/HMM model, which is simply included as baseline with which to compare states with and without inter-feature dependencies.

For the GMM/DBN, we include Scheme A.2, which has the addition of all-parameter embedded training of Phase 3. During Phase 3 training, feature transitions are not forced to match those found in the phone-derived feature labels, and combinations which do not appear in the training data may occur. However, as discussed above, each sub-combination $F_t^k \cup Pa(F_t^k), k = 1, \ldots, 6$ *must* occur in the original phone-derived labelling. Scheme A.2 is primarily included to separate the benefit of embedded training of the GMM observation process parameters from that of learning asynchronous CPTs in Phase 2.

4.4.2 Training scheme B

The Scheme B.1 model has the same observation process parameters as the A.1 model, though uses asynchronous CPTs as learned in Phase 2. The model now has the potential to model feature combinations other than those observed in the training data. Phase 3 all-parameter embedded training of the model in B.1 gives training scheme B.2.

4.5 Experimental results

Early feature recognition experiments revealed numerous insertion errors. A transition penalty was therefore included in all models to balance insertions and deletions, and its value set on held-out validation data. We present results for GMM/DBNs, ANN/HMMs and ANN/DBNs in Sections 4.5.1, 4.5.2 and 4.5.3 respectively.

4.5.1 GMM/DBN feature recognition

Table 9 presents a summary of the results for each of the training schemes listed in Table 8. Test-set recognition results for training Scheme A.1 show that across all 6 features, 86.2% of frames were classified correctly and 79.4% frames were all correct together. The overall feature recognition accuracy was 83.4%. There were 55 feature value combinations found in the output, of which just over half, 28, are feature value combinations which occur in the phone-derived feature value labels.

GMM/DBN				
training	frame-level classification		recognition	feature
scheme	average correct	all correct together	accuracy	combinations
A.1	86.2%	79.4%	83.4%	55
A.2	85.8%	78.6%	83.4%	42
B.1	86.1%	78.8%	83.7%	111
B.2	85.8%	78.0%	83.7%	117

Table 9
AF classification and recognition results for the GMM/DBN model under the different training schemes.

Training Scheme A.2 serves as a control to check that any improvement in the asynchronous models is not simply due to embedded training of the GMMs. However, this was not found to be the case, as there was no increase over the Scheme A recognition accuracy of 83.4%, at the same time as reductions in framewise accuracy.

Early experiments showed that inference speeds for Phase 2 embedded training of feature CPTs were prohibitively slow, so in order to reduce computation, a smaller set of observation process GMMs was used. The models on which training Scheme A.1 results are reported use the set of GMMs found at completion of the mixture component splitting/vanishing regime. These give results of 86.2% average frame-level accuracy and 83.4% overall recognition accuracy, and include 69.6K Gaussian components. A set of GMMs from an intermediate stage of the splitting/vanishing process gave 85.1% frame-level accuracy and 81.1% recognition accuracy using just under 14% of the Gaussian components. These results along with the numbers of Gaussian components are shown in Table 10 where intermediate and final refer to the stage of the splitting/vanishing regime that the GMM sets were taken. The performances of these two systems were considered close enough that the intermediate model parameters could be used to accelerate training the asynchronous CPTs used in training Schemes B.1 and B.2.

The model trained using Scheme B.1, in which asynchronous CPTs are learned, gives a recognition accuracy of 83.7%, which represents a slight increase over that of the A schemes. Furthermore, we observe a greater number of feature combina-

GMM/DBN				
A.1 training stage	frame-level classification		recognition accuracy	# Gaussian components
	average correct	all correct together		
intermediate	85.1%	77.1%	81.1%	9.5K
final	86.2%	79.4%	83.4%	69.6K

Table 10
Close to highest AF recognition results are given by an intermediate model set using substantially fewer Gaussians.

tions in the decoded output, 111, compared to 55 after Scheme A.1. The overall recognition accuracy of 83.7% is identical to that in B.1, with a slight increase in the number of combinations.

Generally, the embedded training schemes (B.1 and B.2) result in a small decrease in the number of frames in which all features are correct together, with B.2 being slightly worse than B.1. These results are as expected: we are observing the model diverging from the phone-derived feature value labels. Comparing the B results with the A results, we can see that allowing non-canonical-phone feature combinations results in a small increase in recognition accuracy. In addition, the number of feature value combinations found in the recognition output is increased by a factor of between 2 and 3. There are 117 combinations found in the final system, compared to 61 in the training data. This suggests that by reinforcing the probability of feature value combinations with strong acoustic likelihood, the system has learned combinations which do not occur in the original phone-derived labelling.

4.5.2 ANN/HMM feature recognition

Results presented in Table 11 show that the overall feature recognition accuracy for the ANN/HMM model is 83.5%, which is very close to the accuracies found for the GMM/DBN. However, compared with the GMM/DBN, an order of magnitude more combinations were found in the ANN/HMM decoded output.

ANN/HMM				
training scheme	frame-level classification		recognition accuracy	feature combinations
	average correct	all correct together		
A.1	86.7%	71.7%	83.5%	3751

Table 11
AF classification and recognition results for the ANN/HMM model for training scheme A.1.

4.5.3 ANN/DBN feature recognition

Table 12 gives a summary of the results found using ANN/DBNs. The recognition accuracy from training Scheme A.1 is 87.8%, which is significant increase over the 83.5% found for the ANN/HMM and reported in Table 11. These models differ only in that the ANN/DBN includes a model of inter-feature dependencies, and we therefore conclude that state-level coupling of features is indeed beneficial to modelling.

ANN/DBN				
training	frame-level classification		recognition	feature
scheme	average correct	all correct together	accuracy	combinations
A.1	89.1%	84.6%	87.8%	54
B.1	89.1%	84.2%	87.8%	97
B.2	88.8%	84.3%	87.8%	83

Table 12
AF classification and recognition results for the ANN/DBN model under the different training schemes.

We find no difference in recognition accuracy between the three training schemes, though the number of feature combinations increases from 54 to 97 after the asynchronous feature CPTs have been estimated in scheme B.1. These results suggest that by using the asynchronous CPTs, the Viterbi (most likely) feature state sequence is modified with respect to the timings of the feature value transitions, rather than the values taken.

There is a decrease in the number of feature combinations found in the decoded output after the observation process ANNs have been retrained on the realigned data for scheme B.2. Of the 83 combinations found in the B.2 decoded output, 23 do not occur within the output of B.1, and 35 do not occur within the feature combinations output by the A.1 model. Even though the overall accuracy is constant, there is clearly some variation within the decoded output, which suggests that the internal structure of the model has diverged from the initial model trained on phone-derived feature labels.

5 Analysis of feature realignment

We investigated whether the embedded training algorithm actually changed the feature value transition times (i.e., shifted the boundaries between two regions which have differing values for that feature), how many changes occur, and where they occur. We performed forced alignment on the training set using the GMM/DBN

to produce an asynchronous feature value labelling of the data and then compared these transcriptions to the phone-derived transcriptions. In the forced alignment, the correct (according to the canonical transcriptions) sequence of features was enforced and silence was synchronized with a silence/non-silence RV which is parent to all feature RVs.

5.1 Overall number of boundary shifts

Table 13 shows both the percentage of frames that differ between the phone-derived and asynchronous feature transcriptions and the number of boundaries that have shifted. Close to half of all boundaries have moved, resulting in changes to 2.4% of all frame-level feature value labels.

feature	% changes	
	boundaries	values
manner	48.7%	2.7%
place	46.2%	3.0%
voicing	49.9%	1.9%
rounding	48.5%	2.5%
front-back	47.2%	2.4%
static	44.5%	1.9%
overall	47.5%	2.4%

Table 13
Percentage of feature transitions and values changed after realigning using DBNs. Results are given per feature and correspond to the training set.

Figure 6 shows the distribution of shifts in feature value transitions, up to 5 frames either side of their locations in the original phone-derived feature value labels. 93.6% of the feature transitions occur within 5 frames of the phone-derived location and 79.0% of transitions are within 1 frame.

5.2 Specific transition shifts

To discover whether linguistically-plausible processes are being captured, we now examine the boundary shifts for individual features. Table 14 lists those feature boundaries that occur in the forced alignment at least 1.5 frames earlier or later than the original phone-derived boundary and how often they occur.

A t-test was used to test whether the deviation from a mean of zero was statistically

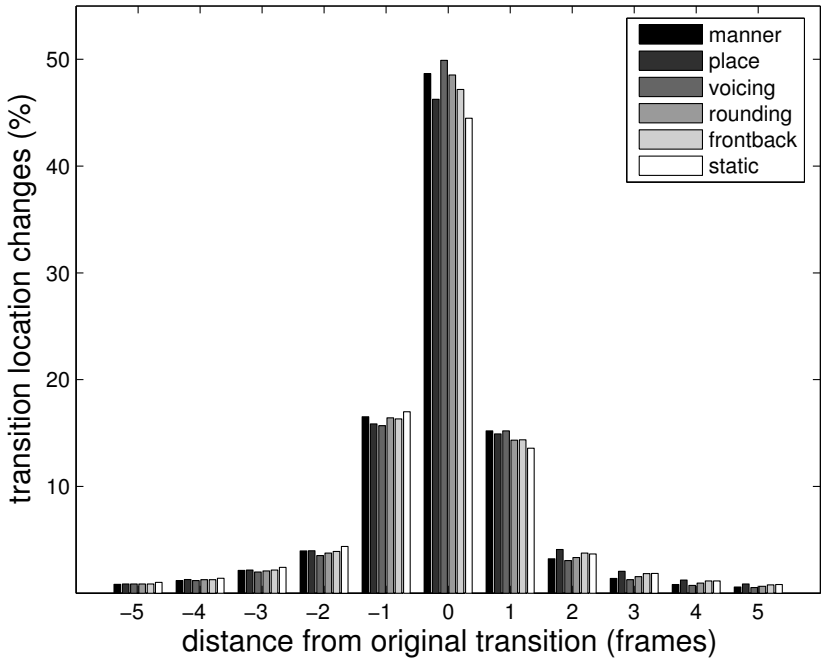

Fig. 6. *Bar graph showing the numbers of frames by which feature boundaries have shifted.*

significant. The last column shows the significance level at which we are able to reject the null hypothesis that the mean deviation is zero. The symbol - is used where the result is not statistically significant.

Because the asynchronous training algorithm does not yet allow deletions, insertions or substitutions of feature labels, there are only a limited number of linguistic processes which we can expect to see in the re-aligned labels. One such process is vowel nasalization. This can occur when a vowel is followed by a nasal consonant, for example, in the words "nine", "one", "and". The expectation is that the vowel-nasal boundary for the *manner* feature will shift to the left, i.e. the end of the vowel will become nasalized. From the data we find that the overall mean deviation for the feature switch "vowel $\rightarrow$ nasal" is -0.24 frames, indicating only slight movement of the nasal feature into the vowel feature. This result is statistically significant at the 1% level which shows that there is a slight but consistent nasal spread into preceding vowels. However, this effect might be reduced as nasal spread is already partially included in the phone-derived feature labels, due to the nasalization diacritic.

An ad-hoc estimate of the degree of asynchrony can be given by the number of feature combinations. The realigned data contains 234 combinations which, compared to the 61 feature combinations in the canonical data, shows that there is indeed asynchronous feature boundary movement.

feature	mean	boundary	count	significance level
manner	-1.62	silence | vowel	4124	1%
	-1.79	stop | silence	1551	1%
	-1.53	fricative | nasal	1561	1%
	-1.74	vowel | stop	133	2.5%
place	-1.90	high | velar	1570	1%
	-1.99	high | mid	1052	1%
	1.83	mid | silence	981	-
	-8.96	silence | high	500	1%
	3.82	alveolar | dental	278	1%
	2.85	low | silence	82	-
voicing	-1.77	silence | voiced	11296	1%
	-1.60	unvoiced | silence	4487	1%
rounding	-2.03	unrounded | rounded	6132	1%
	-1.67	silence | unrounded	4026	1%
front-back	-1.98	central | back	6124	1%
	-2.50	silence | front	2572	1%
	1.76	central | silence	1022	-

Table 14
Mean deviation from canonical boundary for specific feature switches > 1.5 *or* < -1.5 *frames.*

6 Discussion and future work

Previous work on AF recognition by ourselves and others has generally relied completely on phone-derived feature value labels for training. The resulting models inevitably embody many of the "beads-on-a-string" limitations that a feature-based representation is designed to avoid.

In this study, we have moved away from our dependence on phone-derived feature value labels, by using a DBN model in conjunction with an embedded training scheme designed to learn asynchronous feature value changes where supported in the data. The experiments show that this method works. For the GMM/DBN models, training asynchronous models led to a slight increase in feature recognition accuracy from 83.4% to 83.7%, whilst increasing the number of feature combinations found in the output from 55 to 117. Additionally, experiments including

diacritics in the phone-feature mapping show that a more detailed manual labeling of the data does not lead to improvements in accuracy, supporting the case for a data-driven approach.

ANNs currently represent the state-of-the-art in AF recognition, and in this study we have shown that by incorporating a state-level modelling of inter-feature dependencies, we can significantly increase the recognition accuracy. Comparing the ANN/HMM and ANN/DBN results in Tables 11 and 12 respectively, we find that the recognition accuracy is increased from 83.5% to 87.8%, which represents a 26% relative error reduction. This is accompanied by a reduction in the number of feature combinations found in the decoded output from 3751 to 97.

We consider that the order of magnitude fewer feature value combinations produced by the DBNs indicates a suitable operating point between all possible feature value combinations (linguistically implausible) and only those combinations corresponding to canonical phonemes (back to the "beads-on-a-string" problem).

Analysis of the asynchronous feature changes proved to be illuminating: the fact that the model is "transparent" and can be analyzed in such a way is an added benefit of the articulatory feature representation.

Ongoing work includes building a feature-based word recognition system using DBNs. Our choice of DBNs in the current work was made because they offer an ideal framework in which to combine the various components of such a system. The articulatory feature recognition described in this work will form the observation process, with words generated in terms of feature sequences. Because DBNs explicitly represent everything (e.g. the pronunciation dictionary and language model) as model structure (as opposed to special-purpose code in the software), it will be straightforward to re-use the current AF recognition DBN and simply add on the word-to-feature model structure.

Our proposed system will not map via phones at any stage, thus avoiding reintroducing the "beads-on-a-string" paradigm which the feature approach is designed to circumvent. Furthermore, by modelling pronunciation variation in terms of features, we will be able to allow feature insertions, deletions and substitutions during training and decoding; this will increase the power of the embedded training method introduced in this article. We also plan to exploit the fact that accurate recognition of certain features may be more important at some times than others to build a noise-robust system.

Acknowledgements

Many thanks to Jeff Bilmes and his team for GMTK and related advice, also to the anonymous reviewers for their constructive comments.

References

Bilmes, J., October 2002a. GMTK: The Graphical Models Toolkit.
URL `http://ssli.ee.washington.edu/~bilmes/gmtk/`

Bilmes, J., October 2002b. GMTK: The Graphical Models Toolkit. SSLI Laboratory, University of Washington.

Bilmes, J., 2003. Mathematical Foundations of Speech and Language Processing. Institute of Mathematical Analysis Volumes in Mathematics Series. Springer-Verlag, Ch. Graphical Models and Automatic Speech Recognition.

Browman, C., Goldstein, L., 1992. Articulatory phonology: an overview. Phonetica 49, 155–180.

Chang, S., 2002. A syllable, articulatory-feature, and stress-accent model of speech recognition. Ph.D. thesis, University of California, Berkeley, CA.

Chang, S., Greenberg, S., Wester, M., 2001. An elitist approach to articulatory-acoustic feature classification. In: Proceedings of Eurospeech. Aalborg, Denmark, pp. 1725–1728.

Chomsky, N., Halle, M., 1968. The Sound Pattern of English. Harper & Row, New York, NY.

Cole, R., Noel, M., Lander, T., Durham, T., 1995. New telephone speech corpora at CSLU. In: Proceedings of the Fourth European Conference on Speech Communication and Technology. Vol. 1. pp. 821–824.

Eide, E., 2001. Distinctive features for use in an automatic speech recognition system. In: Proceedings of Eurospeech. Aalborg, Denmark, pp. 1613–1616.

Frankel, J., 2003. Linear dynamic models for automatic speech recognition. Ph.D. thesis, The Centre for Speech Technology Research, University of Edinburgh, Edinburgh, UK.

Frankel, J., King, S., 2005. A hybrid ANN/DBN approach to articulatory feature recognition. In: Proceedings of Eurospeech. Lisbon, Portugal, CD-ROM.

Frankel, J., Wester, M., King, S., 2004. Articulatory feature recognition using dynamic Bayesian networks. In: Proceedings of the International Conference on Spoken Language Processing. Jeju, Korea, CD-ROM.

Hacioglu, K., Pellom, B., Ward, W., 2004. Parsing speech into articulatory events. In: Proc. of ICASSP '04. Montreal.

Jensen, F. V., 2001. Bayesian Networks and Decision Graphs. Springer.

Juneja, A., Espy-Wilson, C., 2003. Speech segmentation using probabilistic phonetic feature hierarchy and support vector machines. In: Proceedings of the International Joint Conference on Neural Networks. Portland, Oregon.

Jurafsky, D., Ward, W., Jianping, Z., Herold, K., Xiuyang, Y., Sen, Z., 2001. What kind of pronunciation variation is hard for triphones to model? In: Proceedings of the International Conference on Acoustics, Speech, and Signal Processing. Vol. 1. pp. 577–580.

King, S., Taylor, P., October 2000. Detection of phonological features in continuous speech using neural networks. Computer Speech and Language 14 (4), 333–353.

Kirchhoff, K., 1999. Robust speech recognition using articulatory information. Ph.D. thesis, University of Bielefeld, Bielefeld, Germany.

Lander, T., 15 May 1997. The CSLU labeling guide. Website, `http://www.cslu.ogi.edu/corpora/docs/labeling.pdf`.

Livescu, K., Glass, J., Bilmes, J., 2003. Hidden feature modeling for speech recognition using dynamic Bayesian networks. In: Proceedings of Eurospeech. Vol. 4. Geneva, Switzerland, pp. 2529–2532.

Mariéthoz, J., Bengio, S., 2004. A new speech recognition baseline system for Numbers 95 Version 1.3 based on Torch. Tech. Rep. IDIAP-RR 04-16, IDIAP.

Markov, K., Dang, J., Iizuka, Y., Nakamura, S., 2003. Hybrid HMM/BN ASR system integrating spectrum and articulatory features. In: Proceedings of Eurospeech. Vol. 2. pp. 965–968.

Metze, F., Waibel, A., 2002. A flexible stream architecture for ASR using articulatory features. In: Proceedings of the International Conference on Spoken Language Processing. Denver, CO, CD-ROM.

Morgan, N., Bourlard, H., May 1995. Neural networks for statistical recognition of continuous speech. Proceedings of the IEEE 83 (5), 741–770.

Nix, D., Hogden, J., 1998. Maximum-likelihood continuity mapping (MALCOM): An alternative to HMMs. In: Kearns, M., Solla, S., Cohn, D. (Eds.), Proceedings of the Advances in Neural Information Processing Systems Conference, NIPS. Vol. 11. MIT Press, pp. 744–750.

Niyogi, P., Burges, C., Ramesh, P., 1999. Distinctive feature detection using support vector machines. In: Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-99). Phoenix, AZ.

Ostendorf, M., 1999. Moving beyond the 'beads-on-a-string' model of speech. In: Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop. Vol. 1. Keystone, Colorado, USA, pp. 79–83.

Renals, S., Hochberg, M., 1995. Efficient search using posterior phone probability estimators. In: Proceedings of the International Conference on Acoustics, Speech, and Signal Processing. Detroit, MI., pp. 596–599.

Richards, H. B., Bridle, J. S., 1999. The HDM: A segmental hidden dynamic model of coarticulation. In: Proceedings of the International Conference on Acoustics, Speech, and Signal Processing. Vol. 1. Phoenix, AZ, pp. 357–360.

Robinson, A., Cook, G., Ellis, D., Fosler-Lussier, E., Renals, S., Williams, D., 2002. Connectionist speech recognition of broadcast news. Speech Communication 37, 27–45.

Scharenborg, S., Wan, V., Moore, R., 2006. Capturing fine-phonetic variation in speech through automatic classification of articulatory features. In: Proceedings of the workshop on Speech Recognition and Intrinsic Variation. Toulouse,

France, pp. 77–82.

Stephenson, T., 1998. Speech recognition using phonetically featured syllables. Master's thesis, University of Edinburgh.

Stephenson, T., Bourlard, H., Bengio, S., Morris, A., 2000. Automatic speech recognition using dynamic Bayesian networks with both acoustic and articulatory variables. Tech. Rep. 00-19, IDIAP.

Ström, N., 1997. Phoneme probability estimation with dynamic sparsely connected artificial neural networks. The Free Speech Journal Issue #5.

Wester, M., 2003. Syllable classification using articulatory-acoustic features. In: Proceedings of Eurospeech. Geneva.

Wester, M., Frankel, J., King, S., 2004. Asynchronous articulatory feature recognition using dynamic Bayesian networks. In: Proceedings of the Institute of Electronics, Information and Communication Engineers Beyond HMM Workshop. Vol. 104. Kyoto, Japan, pp. 37–42, SP2004-81-95.

Wester, M., Greenberg, S., Chang, S., 2001. A Dutch treatment of an elitist approach to articulatory-acoustic feature classification. In: Proceedings of Eurospeech. Aalborg, Denmark, pp. 1729–1732.

Wrench, A. A., 2001. A new resource for production modelling in speech technology. In: Proceedings of the Workshop on Innovations in Speech Processing. Stratford-upon-Avon, UK.

Young, S., Evermann, G., Kershaw, D., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., Woodland, P., 2002. The HTK Book (for HTK Version 3.2). Cambridge University Engineering Department, Cambridge, UK.

Zlokarnik, I., May 1995. Adding articulatory features to acoustic features for automatic speech recognition. The Journal of the Acoustical Society of America 97 (5), 3246.

Appendix

A Articulatory acoustic phone to feature specification.

This table gives the phone to feature mapping for all phones that occur in OGI Numbers. The examples are taken from Lander (1997). Phones with diacritics are included in the list but no examples are given. Abbreviations are explained in A.2.

		feature groups					
Worldbet phones	example	manner	place	voice	front-back	round	static
9r, l, l=	**r**ight, **l**ight	approximant	alveolar	+voice	nil	nil	dyn
&r, 3r	butt**er**, b**ir**d	approximant	mid	+voice	central	-round	dyn
j, w	**y**et, **wh**en	approximant	velar	+voice	nil	nil	dyn
9r_0, 9r_h_0,		approximant	alveolar	-voice	nil	nil	dyn
w_0, w_0_h		approximant	velar	-voice	nil	nil	dyn
f, v_0	**f**ine	fricative	labdent	-voice	nil	nil	static
v	**v**ine	fricative	labdent	+voice	nil	nil	static
tS, th_F	**ch**urch	fricative	alveolar	-voice	nil	nil	dyn
s, tc_F, z_0,	**s**ign	fricative	alveolar	-voice	nil	nil	static
z	re**s**ign	fricative	alveolar	+voice	nil	nil	static
T	**th**igh	fricative	dental	-voice	nil	nil	static
T_v, T_v_r		fricative	dental	+voice	nil	nil	static
T_w,		fricative	dental	-voice	nil	+round	static
kh_F		fricative	velar	-voice	nil	nil	dyn
h, kc_F,	**h**ope	fricative	velar	-voice	nil	nil	static
h_v		fricative	velar	+voice	nil	nil	static
m, m_(	**m**e	nasal	labdent	+voice	nil	nil	static
n	**kn**ee	nasal	alveolar	+voice	nil	nil	static
n=	butt**on**	nasal	alveolar	+voice	nil	nil	dyn
n_0		nasal	alveolar	-voice	nil	nil	static
i:_~		nasal	high	+voice	front	-round	dyn

continued from previous page							
Worldbet phones	example	manner	place	voice	front-back	round	static
A_~		nasal	low	+voice	back	+round	static
aI_~		nasal	low_high	+voice	cnt_fr	-round	dyn
^_?_; ^_; ^_~_?		nasal	mid	+voice	central	-round	static
E_~		nasal	mid	+voice	front	-round	static
h_~		nasal	velar	-voice	nil	nil	static
w_;		nasal	velar	+voice	nil	nil	dyn
N_(, N_?		nasal	velar	+voice	nil	nil	static
b, b_(	**b**an	stop	labdent	+voice	nil	nil	dyn
d, th_(_v, th_v	**d**an	stop	alveolar	+voice	nil	nil	dyn
g	**g**ander	stop	velar	+voice	nil	nil	dyn
th	**t**an	stop	alveolar	-voice	nil	nil	dyn
th_w		stop	alveolar	-voice	nil	+round	dyn
kh	**c**an	stop	velar	-voice	nil	nil	dyn
u	b**oo**t	vowel	high	+voice	back	+round	dyn
U	b**oo**k	vowel	high	+voice	back	+round	static
i:	b**ee**t	vowel	high	+voice	front	-round	dyn
I	b**i**t	vowel	high	+voice	front	-round	static
>, A	c**au**ght, f**a**ther	vowel	low	+voice	back	+round	static
@, @_?	b**a**t	vowel	low	+voice	front	-round	static
aU	ab**ou**t	vowel	low_high	+voice	cnt_back	-rnd_+rnd	dyn
aI	b**ye**	vowel	low_high	+voice	cnt_fr	-round	dyn
>i	b**oy**	vowel	low_mid	+voice	back_fr	+rnd_-rnd	dyn
&,^ E_x	**a**bove, ab**o**ve	vowel	mid	+voice	central	-round	static
E	b**e**t	vowel	mid	+voice	front	-round	static
oU	b**oa**t	vowel	mid_high	+voice	cnt_back	-rnd_+rnd	dyn
ei	b**a**y	vowel	mid_high	+voice	front	-round	dyn
&_0, ^_0		vowel	mid	-voice	central	-round	static
u_0,		vowel	high	-voice	back	+round	dyn

continued from previous page							
Worldbet phones	example	manner	place	voice	front-back	round	static
i:_0		vowel	high	-voice	front	-round	dyn
@_0		vowel	low	-voice	front	-round	static
aI_0		vowel	low_high	-voice	cnt_fr	-round	dyn
E_0		vowel	mid	-voice	front	-round	static
oU_0		vowel	mid_high	-voice	cnt_back	-rnd_+rnd	dyn
ei_0		vowel	mid_high	-voice	front	-round	dyn
ei_?_w		vowel	mid_high	+voice	front	+round	dyn
oU_w		vowel	mid_high	+voice	cnt_back	+round	dyn
E_w		vowel	mid	+voice	front	+round	static
i:_w,		vowel	high	+voice	front	+round	dyn
^_?_w, ^_w		vowel	mid	+voice	central	+round	static
I_x		vowel	high	+voice	central	-round	static
u_x		vowel	high	+voice	central	+round	dyn
A_x		vowel	low	+voice	central	+round	static
aI_x_?		vowel	low_high	+voice	central	-round	dyn
dc,tc,tSc,	closure	sil	sil	sil	sil	sil	sil
kc, kc_v +	closure	sil	sil	sil	sil	sil	sil
pau	pause	sil	sil	sil	sil	sil	sil

Table A.1: *AF specifications.*

Abbreviations used in Table A.1	
sil	silence
approx	approximant
labdent	labiodental
cnt	central
fr	front
+rnd	round
-rnd	unround
dyn	dynamic

Table A.2
Abbreviations used in Table A.1